Utilizing an Island Model for EA to Preserve Solution Diversity for Inferring Gene Regulatory Networks

Christian Spieth, Felix Streichert, Nora Speer, and Andreas Zell
Centre for Bioinformatics Tübingen (ZBIT)
University of Tübingen
D-72076 Tübingen, Germany
Email: spieth@informatik.uni-tuebingen.de

***Abstract*— In this paper we address the problem of finding gene regulatory networks from artificial data sets of DNA microarray experiments. Some researchers suggested Evolutionary Algorithms for this purpose. We suggest to use an enhancement for Evolutionary Algorithms to infer the parameters of the non-linear system given by the observed data more reliably and precisely. At present, we use S-Systems because they are a general mathematical model for simulating the complex interactions of gene regulatory networks. Due to the limited number of available data, the inferring problem is highly under-determined and ambiguous. Further on, the problem often is highly multi-modal and therefore appropriate optimization strategies become necessary. We propose to use an island model to maintain diversity in the EA population to prevent premature convergence and to raise the probability of finding the global optimum.**

I. INTRODUCTION

Fifty years ago, Watson and Crick identified the physical structure of the DNA, thus starting a new age for biological research. Their discovery made it possible to describe diseases on a commonly agreed theoretical base. Since then, Systems Biology has become an important field in biology, which aims at deep insights into biological systems. While new techniques in molecular genetics for measuring gene expression levels of a single cell or tissue like DNA microarrays led to remarkable advances in the understanding of processes at the system level of an organism, the main focus of current research is mainly on the identification of genes that show significant changes between different experimental conditions, or which can be clustered due to the same course of expression over time. The next step is to understand the principles of biological systems grounded on the molecular level. To provide a deep understanding of life, we have to understand not only the components of the systems but also their dependencies, interactions, and structures.

Within the last few years, researchers obtained large amounts of data sets of gene expression experiments, which were mostly analyzed with standard low-level analysis methods like differential comparison, clustering or classification. But these methods are using only a very small part of the information hidden in the data sets. A system-level approach is necessary, if we want to incorporate large amounts of data into a comprehensive model of the structure and functions of the complex mechanisms within an organism.

Recently developed DNA microarray technology allows measurement of gene expression levels for a whole genome at the same time. Experiments using this technique provide new insights into activities of genes under different biochemical and physiological environment conditions and can therefore be used to extract time-dependent relationship information of interacting genes, i.e. to identify gene regulatory networks. A gene regulatory network (GRN) defines the complicated structure of dependencies of RNA produced by one expressed gene influencing regulatory mechanisms of other genes. The amount of expression data grows rapidly because this technique allows for high-throughput experiments. And although increasing numbers of microarray data sets become available, mathematical methods are infeasible to determine regulatory networks from a small number of chips. Several approaches have been suggested in the past few years addressing this problem. The following section will give an overview over techniques to infer gene regulatory networks.

A. Related Work

Inferring the underlying relationships between genes is subject to current research and has recently become one of the major topics in bioinformatics due to the increased computing power available. There have been some approaches in the field of Systems Biology to solve the combinatorial problem of the inference process.

The earliest models to simulate regulatory systems found in the literature are Boolean or Random Boolean Networks (RBN) [15]. In Boolean Networks gene expression levels can be in one of two states: either 1 (on) or 0 (off). The quantitative level of expression is not considered. Two examples for inferring Boolean Networks are given by Akutsu [1] and the REVEAL algorithm by Liang *et al.* [19]. These models have the advantage that they can be solved with only small computational effort. But they suffer from the disadvantage of being tied to discrete system states.

In contrast to discrete methods like RBNs, qualitative network models allow for multiple levels of gene regulation as presented for example by Thieffry and Thomas [26]. Akutsu *et al.* [2] suggest a heuristic for inferring such models from time series data.

There are a number of variants for quantitative networks. First, the weighted matrix model by Weaver *et al.* [29], for example, considers the continuous level of gene expression. This approach parameterizes the mathematical model with discrete time and linear relationships between the components of the system. The topology and the parameters of this model have been successfully inferred by the use of Genetic Algorithms in [3] and [4].

An alternative model for GRNs, used to infer regulatory mechanisms, are S-Systems [23]. S-Systems have recently become popular. They have been examined in several publications. Tominaga *et al.* [27], for example, inferred S-Systems by using a real-coded GA. They examine two examples with $N = 2$ and $N = 5$ genes, respectively, but only selected genes were inferred in the 5-dimensional regulatory network. Kikuchi *et al.* [16] used the same approach as Tominaga, but introduced a modification to gain sparse matrices.

Further on, differential equations can be used to describe the interactions between genes in a regulatory system. Chen *et al.* [9] and de Hoon *et al.* [10] introduced methods to find the parameters of linear differential equations by using specialized heuristics.

Examples for non-parameterized quantitative networks are arbitrary differential equations, which are the most flexible and powerful approach to model the dependencies of genes. The most prominent method to work on arbitrary differential equation systems is Genetic Programming (GP) as shown for reverse engineering of pathways by Koza [18] and for regulatory networks by Ando *et al.* [5] and Sakamoto and Iba [22].

B. Motivation

The methods using EAs suggested in the literature face several problems. One of them is that the EA often converges prematurely to local optima. Due to the deceptiveness and multi-modality of the search space, it is very likely that even with repeated runs of the optimization process only more local optima are found. Thus, even a multi-run optimizing process results in only suboptimal network models. To bypass this issue, we use an island strategy to preserve the diversity in the EA population and thus increase the probability of finding better solution than the standard algorithms.

The following publication is structured as follows. Detailed description of our proposed method will be given in section II and III. Applications and results will be shown in sections IV. Finally, conclusions and an outlook on future research will be covered by sections V and VI.

II. MATHEMATICAL MODEL

On an abstract level, the behavior of a cell under given environmental conditions is represented by gene regulatory dependencies of N genes, where N is either the number of genes in the genome or the number of genes in a specific sub-network, e.g. the immune pathway. Each of these genes g_i produces a certain amount of RNA x_i when expressed. It is known that RNA or RNA products may induce the activation of other genes. Therefore, the overall concentration of the RNA changes over time depending on the concentrations of other RNA levels: $\vec{x}(t+1) = h_i(\vec{x}(t))\,,\ \vec{x}(t) = (x_1, \cdots, x_n)$, where h_i describes the change of each RNA level depending on all or only on some RNA concentrations in the previous time step.

To model and to simulate regulatory networks we decided to use S-Systems since they are well-documented and examined and are flexible.

1) S-Systems: S-Systems are a type of power-law formalism, which has been suggested by Irvine and Savageau [23], [14]. S-Systems are systems of differential equations, which have been derived from a Taylor approximation of a system of arbitrary differential equations.

They are given by a set of nonlinear differential equations as follows:

$$\frac{dx_i(t)}{dt} = \underbrace{\alpha_i \prod_{j=1}^{N} x_j(t)^{\mathcal{G}_{i,j}}}_{\text{synthesis}} - \underbrace{\beta_i \prod_{j=1}^{N} x_j(t)^{\mathcal{H}_{i,j}}}_{\text{degradation}} \tag{1}$$

where $\mathcal{G}_{i,j}$ and $\mathcal{H}_{i,j}$ are kinetic exponents, α_i and β_i are positive rate constants and N is the number of equations in the system. The equations in (1) can be seen as divided into two components: a synthesizing and a degradation component.

The kinetic exponents $\mathcal{G}_{i,j}$ and $\mathcal{H}_{i,j}$ determine the structure of the regulatory network. In the case $\mathcal{G}_{i,j} > 0$ gene g_j induces the synthesis of gene g_i. If $\mathcal{G}_{i,j} < 0$ gene g_j inhibits the synthesis of gene g_i. Analogously, a positive (negative) value of $\mathcal{H}_{i,j}$ indicates that gene g_j induces (suppresses) the degradation of the mRNA level of gene g_i.

The S-System formalism has a major disadvantage in that it includes a large number of parameters that have to be estimated. The total number of parameters in S-Systems is $2N(N+1)$, with N the number of state variables x_i (genes). This causes problems with increasing number of participating genes due to the quadratically increasing number of parameters to infer. The parameters of the S-System $\vec{\alpha}$, $\vec{\beta}$, $\mathcal{G}$, and $\mathcal{H}$ are optimized with Evolutionary Algorithms described in the following paragraph.

III. OPTIMIZATION TECHNIQUES

The following sections describe the optimization algorithms used in this publication. The following paragraphs will also give a brief introduction to the general principles of Evolutionary Algorithms.

A. Evolutionary Algorithms

Evolutionary Algorithms are stochastic optimization techniques that mimic the natural evolution process of repeated mutation and selection as proposed by Charles Darwin. They have proved to be a powerful tool for solving complex optimization problems and in particular combinatorial problems. Three main types of evolutionary algorithms have been proposed in the last decades: Genetic Algorithms (GA), mainly developed by J.H. Holland [13], Evolution Strategies (ES), developed by I. Rechenberg [21] and H.-P. Schwefel [24], and Genetic Programming (GP) by J.R. Koza [17]. Each of these uses different solution representations and different operators working on them.

1) Genetic Algorithm (GA): Genetic Algorithms imitate the evolutionary processes with emphasis on genotype based operators (genotype/phenotype dualism). The GA works on a population of artificial chromosomes, referred to as individuals. Each individual is represented by a string of L bits. Each segment of this string corresponds to a variable of the optimizing problem in a binary encoded form.

The population is evolved in the optimization process mainly by crossover operations. This operation recombines the bit strings of individuals in the population with a certain probability p_c. Mutation is secondarily in most applications of a GA. It is responsible to ensure that some bits are changed, thus allowing the GA to explore the complete search space even if necessary alleles are temporarily lost due to convergence.

2) Evolution Strategies (ES): The second type of an Evolutionary Algorithm is the Evolution Strategy. ES differ from GAs mainly in respect to the representation of solutions and the selection operators. They mainly rely on sophisticated mutation operators, smaller population sizes and an increased selection pressure.

The selection of the individuals forming a population is deterministic, as in contrast to GAs, where a stochastic method is used. In case of the (μ, λ)-ES selection strategy, the μ best individuals from a population of λ offsprings are selected to create the next population. An alternative implementation is the $(\mu + \lambda)$-strategy, which selects the μ best individuals from the population of the λ offsprings joined with the old population of μ parents.

B. Island Strategy

Island strategies have been suggested as an improvement for EAs for many problem types and they are well documented. An overview on distributed EA and different distribution schemes can be found in [8]. Other applications of island strategies can be found, for example, in [6], [7], and [28]. We suggest to utilize the abilities of island models in the inference process of gene regulatory systems to maintain the diversity within the EA population and to reduce the chance of premature convergence. The general principle of an island strategy is a set of l EA populations, which evolves independently for m generations. Then migration occurs.

```
Island Strategy
begin
   initialize island populations
   while (termination criteria not met)
      evolve populations for m generations
      migrate best individuals
   endwhile
end
```

Fig. 1. General principle of the island strategy

Migration is implemented in our algorithm as selecting the best individuals from each EA population, which are then mutated and recombined to form new island populations. After migration, each EA population evolves independently again. The general principle of an island strategy is outlined in fig. 1. Fig. 2 shows schematically the two phases of the island strategy, i.e. the independent evolution of subpopulations and the migration phase.

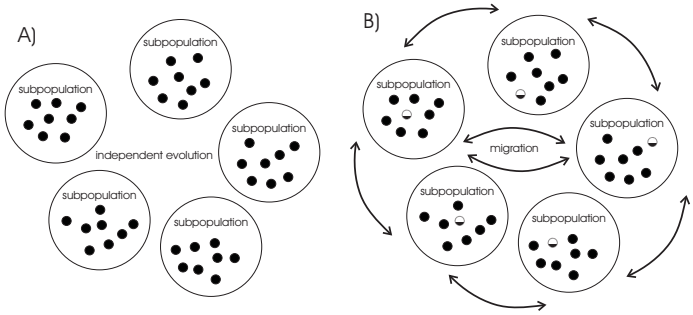

Fig. 2. Two phases of the island model. (A: independent evolution of subpopulations, B: migration)

Having multiple populations initialized independently ensures to have a diverse set of individuals covering a large part of the optimization search space. With migration, good solution elements are able to spread over the l subpopulations. This can enable a subpopulation to escape a local optimum and thus increase the performance of the overall algorithm.

C. Fitness

For assessing the quality of the locally obtained results we use the following equation for calculation of the fitness values for the optimization process:

$$f = \sum_{i=1}^{N} \sum_{k=1}^{T} \left(\frac{\hat{x}_i(t_k) - x_i(t_k)}{x_i(t_k)} \right)^2 \qquad (2)$$

where N is the total number of genes in the regulatory system, T is the number of sampling points taken from the time series and $\hat{x}$ and x distinguish between estimated data and experimental data. The overall problem is to minimize the fitness value f. This function has been used in several publications [16], [20], [27].

IV. APPLICATIONS

Our approach was tested on small artificial gene regulatory networks ($N \leq 20$ genes). To test the method we created artificial microarray data sets, which were to be reverse engineered by the compared algorithms. The data sets were randomly created and simulated. Because GRNs are sparse systems in nature, we created regulatory networks randomly with a maximum cardinality of $k \leq 3$, i.e. each of the N genes depends on three or less other genes within the network.

EA parameters. We compared the island strategy with two standard algorithms, a standard GA and a standard ES. The GA used a population of possible solutions with 500 individuals, tournament selection strategy with a tournament group size of 8 and a 3-point crossover-operator with a crossover probability of $p_c = 1.0$ and a mutation probability of $p_m = 0.1$. The decision variables are binary encoded using 32 bits and one-point mutation was applied onto the genotype.

The inference by the standard ES (real-value encoding) was performed using a (μ,λ)-ES with $\mu = 10$ parents and $\lambda = 100$ offsprings together with a Covariance Matrix Adaptation (CMA) mutation operator [12] without recombination. In case of the ES, the probabilities of crossover and mutation were chosen as $p_c = 0.0$ and $p_m = 1.0$.

In case of the proposed island-strategy, two different implementations were tested. The first used $l = 10$ (μ,λ)-ES island populations with $\mu = 10$, $\lambda = 50$, CMA and no recombination. Migration took place after each ES-island terminated due to a fixed number of fitness evaluations. The second was implemented with a GA with a population size of 100 individuals encoding real values with 32-bits. For recombination a 3-point crossover (crossover probability of $p_c = 1.0$) was used with tournament selection with a tournament group size of 8 and one-point mutation (mutation probability of $p_m = 0.1$).

All optimizations were repeated $m = 20$ times to gain an averaged course of fitness values and the EA settings were determined in preliminary experiments. To compare the results of the three methods, a total number of fitness evaluations $N_{max} = 1,000,000$ was specified.

A. $N = 5$ Genes

The dynamics of this artificial gene regulatory network is shown in fig. 3. The fitness courses for the three methods are given in the following graph (see fig. 4).

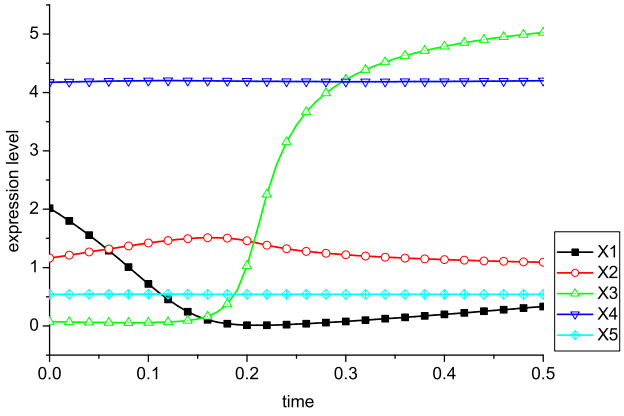

Fig. 3. Artificial gene regulatory system ($N = 5$)

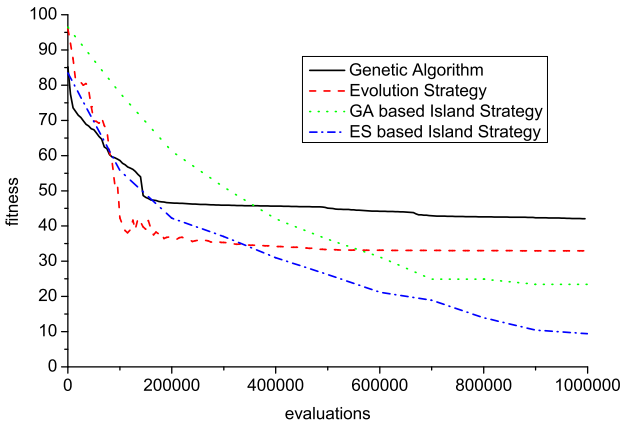

Fig. 4. Fitness of the inference process ($N = 5$)

As can be seen in the graph, the standard GA started with better fitness values due to the larger population size. Further on, the GA optimized until the termination criterion was reached suggesting better results with increased number of total evaluations. The ES converged faster than the GA in the beginning, but started to stagnate after approximately $250,000$ evaluations on average. This is most likely because it cannot escape a local optimum. In contrast to this, both island strategies improved the fitness value continuously and seemed to be not converged at the end of the optimization, which suggests even better results with a larger number of fitness evaluations are possible. The GA based island strategy started with a larger population size and therefore with better fitness values as the ES based island strategy. But during the optimization process, the ES based strategy resulted in better solutions regarding the fitness function than the other algorithms. This implementation used the advantage of the ES to converge faster to optima than a GA. The island strategy ensured that the ES populations do not converge to the very

same subspace of the optimization space, i.e. converge to the same optima.

B. $N = 10$ *Genes*

As a second test case we created another 10-dimensional regulatory network randomly. The dynamics of the example are given in fig. 5.

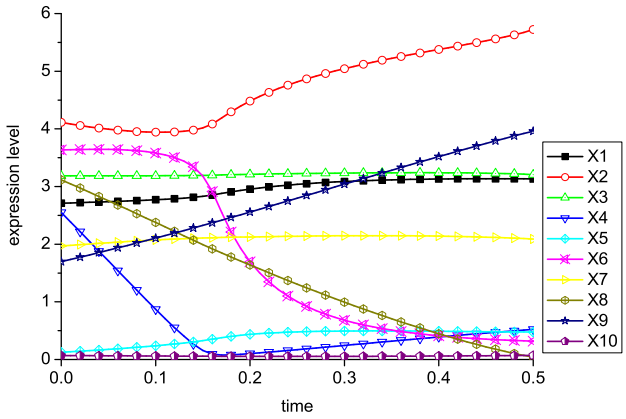

Fig. 5. Artificial gene regulatory system ($N = 10$)

The optimization processes were performed as in the example before, but with a higher number of fitness evaluations. Each algorithm was terminated after a total number of $N_{max} = 2,000,000$ evaluations to pay respect to the increased number of parameters of the model.

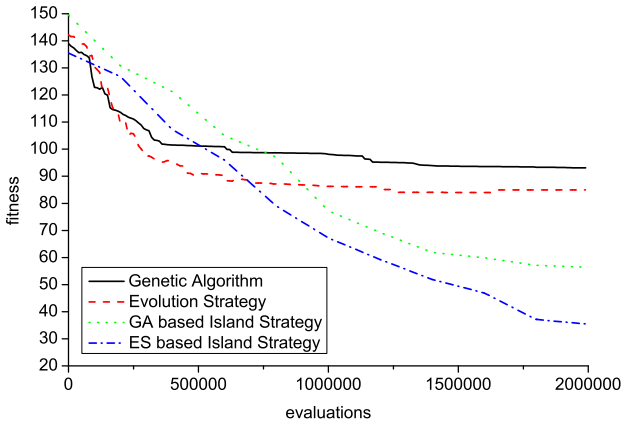

Fig. 6. Fitness of the inference process ($N = 10$)

Fig. 6 shows the fitness course averaged over the 20 repeated optimization runs. As in the example before, both island strategies outperformed the two conventional methods by finding better models with respect to the fitness value. The ES converged again to a local optimum without being able to escape. The standard GA started with better fitness values but was not able to converge to solutions with a comparable good fitness value than the two island strategies. Again, the island strategies performed best with an advantage for the ES based island strategy.

C. $N = 20$ *Genes*

The third GRN inferred with the proposed methods is an artificial 20-dimensional system. The simulated time courses are not given here because the large number of components of the system makes the graph unclear. The optimization was performed with the same parameter settings as described in the previous section (see section IV-B, $N = 10$ genes).

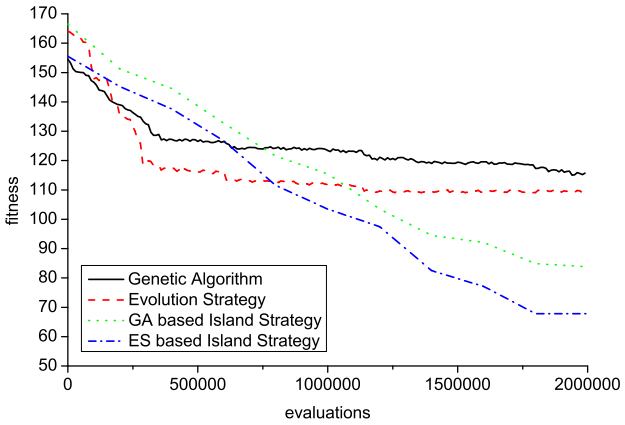

Fig. 7. Fitness of the inference process ($N = 20$)

Fig. 7 shows the averaged fitness course. Again, the standard ES and the GA did not find a solution with the given number of fitness evaluations. The ES converged after approximately $1,200,000$ fitness evaluations to a local optimum and the GA optimizes until the termination criterion was reached. The island strategies converged to optima with good fitness values, suggesting again that they were not fully converged at the end of the optimization process. And again the ES based island strategy converged faster than the GA based.

V. CONCLUSION

The problem of inferring GRNs is a very difficult process due to the limited data available and the large number of unknown variables in the system. One of the problems found in the literature is that conventional methods repeatedly run into local optima, thus being not able to find the optimal solution.

In this paper we suggested to use island strategies to infer gene regulatory networks, because they efficiently preserve the diversity of network candidates in the optimization process of inference problems to find genetic networks from time-series data. We showed that island strategies were able to find better solutions with respect to the fitness than the standard methods. Further on, the proposed methods converged faster to good solutions. The ES based island strategy performed better than the GA based implementation. This is most likely because it

uses the advantage of the ES to converge faster to optima than the GA. Additionally, the island strategy ensures diversity in the subpopulations, thus resulting in better solutions. Therefore, island strategy algorithms show improved quality of results and can be used together with other techniques to clearly identify optimal network models.

Further on, our algorithms proved to work even for middle-sized examples ($N = 20$ genes). Most examples found in literature are artificial and very small, i.e. with a total number of ten genes or lower, while in biological networks even small systems have at least 50–100 components. We showed that our method is able to handle sparse systems ($k \leq 3$) with 20 genes, restricted currently only by computational performance. Future experiments on high performance computers will address large-scale systems with at least 100 genes.

VI. FUTURE WORK

In future work, we will exploit the ability of island models to result in better solutions by combining island strategies with other enhancements of the inferring process. For example, iterative methods [25] can be used to iteratively identify the correct regulatory network model by incorporating additional microarray data sets.

Furthermore, we will continue to test our method with real microarray data in close collaboration with biological researchers at our facility. In future work we plan to use real microarray data sets and to include a-priori information into the inference process like partially known pathways or information about co-regulated genes, which can be found in literature.

Additionally, other models for gene regulatory networks will be examined for simulation of the non-linear interaction system as listed in Section II to overcome the problems with a quadratic number of model parameters of the S-System.

ACKNOWLEDGMENT

This work was partially supported by the National Genome Research Network (NGFN) [11] from the Federal Ministry of Education and Research in Germany.

REFERENCES

[1] T. Akutsu, S. Miyano, and S. Kuhura. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 17–28, 1999.

[2] T. Akutsu, S. Miyano, and S. Kuhura. Algorithms for identifying boolean networks and related biological networks based on matrix multiplication and fingerprint function. In *Proceedings of the fourth annual international conference on Computational molecular biology*, pages 8 – 14, Tokyo, Japan, 2000. ACM Press New York, NY, USA.

[3] S. Ando and H. Iba. Quantitative modeling of gene regulatory network - identifying the network by means of genetic algorithms. In *Poster Session of Genome Informatics Workshop 2000*, pages 278–280, 2000.

[4] S. Ando and H. Iba. Inference of gene regulatory model by genetic algorithms. In *Proceedings of the 2001 Congress on Evolutionary Computation*, pages 712–719. IEEE Press, 2001.

[5] S. Ando and H. Iba. Modeling genetic network by hybrid GP. In *Proceedings of the 2002 Congress on Evolutionary Computation*, pages 291–296. IEEE Press, 2002.

[6] T. Bäck. *Evolutionary algorithms in theory and practice*. Oxford University Press, 1996.

[7] T. Bäck, D. Fogel, and Z. Michalewicz. *Handbook on Evolutionary Computation*. Oxford University Press, 1997.

[8] E. Cantu-Paz. A survey of parallel genetic algorithms. Technical Report 91003, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 1997.

[9] T. Chen, H. L. He, and G. M. Church. Modeling gene expression with differential equations. In *Proceedings of the Pacific Symposium on Biocomputing*, 1999.

[10] M. J. de Hoon, S. Imoto, K. Kobayashi, N. Ogasawara, and S. Miyano. Inferring gene regulatory networks from time-ordered gene expression data of bacillus subtilis using differential equations. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 8, pages 17–28, 2003.

[11] German Federal Ministry of Education and Research (BMBF). National Genome Research Network (NGFN). www.ngfn.de, 2001.

[12] N. Hansen and A. Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *Proceedings of the 1996 IEEE Int. Conf. on Evolutionary Computation*, pages 312–317, Piscataway, NJ, 1996. IEEE Service Center.

[13] J. H. Holland. *Adaption in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Systems*. The University Press of Michigan Press, Ann Arbor, 1975.

[14] D. H. Irvine and M. A. Savageau. Efficient solution of nonlinear ordinary differential equations expressed in S-systems canonical form. *SIAM Journal of Numerical Analysis*, 27(3):704–735, 1990.

[15] S. A. Kauffman. *The Origins of Order*. Oxford University Press, New York, 1993.

[16] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, and M. Tomita. Dynamic modeling of genetic netowrks using genetic algorithm and s-sytem. *Bioinformatics*, 19(5):643–650, 2003.

[17] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.

[18] J. R. Koza, W. Mydlowec, G. Lanza, J. Yu, and M. A. Keane. Reverse engineering of metabolic pathways from observed data using genetic programming. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 6, pages 434–445, 2001.

[19] S. Liang, S. Fuhrman, and R. Somogyi. REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 3, pages 18–29, 1998.

[20] Y. Maki, D. Tominaga, M. Okamoto, S. Watanabe, and Y. Eguchi. Development of a system for the inference of large scale genetic networks. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 6, pages 446–458, 2001.

[21] I. Rechenberg. *Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart, 1973.

[22] E. Sakamoto and H. Iba. Inferring a system of differential equations for a gene regulatory network by using genetic programming. In *Proceedings of the 2001 Congress on Evolutionary Computation*, pages 720–726. IEEE Press, 2001.

[23] M. A. Savageau. 20 years of S-systems. In E. Voit, editor, *Canonical Nonlinear Modeling. S-systems Approach to Understand Complexity*, pages 1–44, New York, 1991. Van Nostrand Reinhold.

[24] H.-P. Schwefel. *Numerical optimization of computer models*. John Wiley and Sons Ltd, 1981.

[25] C. Spieth, F. Streichert, N. Speer, and A. Zell. Iteratively inferring gene regulatory networks with virtual knockout experiments. In R. et al., editor, *Proceedings of the 2nd European Workshop on Evolutionary Bioinformatics (EvoWorkshops 2004)*, volume 3005 of *LNCS*, pages 102–111, 2004.

[26] D. Thieffry and R. Thomas. Qualitative analysis of gene networks. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 77–87, 1998.

[27] D. Tominaga, N. Kog, and M. Okamoto. Efficient numeral optimization technique based on genetic algorithm for inverse problem. In *Proceedings of German Conference on Bioinformatics*, pages 127–140, 1999.

[28] R. K. Ursem. Multinational evolutionary algorithms. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, editors, *Proceedings of the Congress of Evolutionary Computation (CEC99)*, volume 3, pages 1633–1640. IEEE Press, 1999.

[29] D. Weaver, C. Workman, and G. Stormo. Modeling regulatory networks with weight matrices. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 4, pages 112–123, 1999.